From: Rick Hornbeck
To: Microsoft ATR
Date: 1/28/02 11:50pm
Subject: Microsoft Settlement

To Whom It May Concern:

I have attached three documents that explain my position on the Microsoft Antitrust Case, along with a proposed solution. I also attached a copy of my resume to establish my credibility and to assist you in determining the value that you should place on my recommendations.

I do not believe that the Technical Committee will have the necessary access to key Microsoft personnel or the enforcement authority, either directly or indirectly, to make a difference.

Although the Proposed Settlement contains several good measures for curtailing some of Microsoft's anti-competitive actions, it does not go far enough.

I believe that the solution I propose in the first attached document will level the playing field to the degree needed to make a long-term positive impact.

Regards,

Rick Hornbeck
Hornbeck Consulting
556 S. Fair Oaks Ave., Suite 346
Pasadena, CA 91105
Rick_Hornbeck@pacbell.net
(cell) +1 323 363-2151
(efax) +1 208 275-1245

January 28, 2002

U.S. Department of Justice
Anti-trust Division
Microsoft.atr@usdoj.gov

To Whom It May Concern:

I am writing to convey my proposed solution to the Microsoft anti-trust case.

The dilemma is how to prevent Microsoft from using its monopolistic power in the future, to weaken competition, consumer choice, and innovation.

Breakup along product lines is problematic due to Microsoft's successful public relations disinformation campaign

Microsoft has astutely intertwined its various products so tightly that any breakup of the corporation is unrealistic, if it occurs along product lines, requiring each new organization to become its own, independent profit center. At least that is what Microsoft would have us believe.

Although such a restructuring is possible, Microsoft's argument would be that it would reduce the value of each product by approximately 30% because it would eliminate the benefits derived from their capability to 'interconnect,' and exchange data 'seamlessly.'

In my opinion, this interconnectivity argument is flawed, as virtually the same quantity and quality of interconnectivity has existed amongst Microsoft's products for many years.

Microsoft is notorious for inflating the value of its product's features in the media, in advertising, and in supposedly objective articles written by shills in technical journals. However, it has failed to introduce significant new interconnectivity feature enhancements over the past few years, and it is unlikely that any new advancements or features in this area are forthcoming.

In addition, the other major vendors in the desktop software market already offer the same level of interconnectivity between their own products and Microsoft's products, in the only area that really matters – cutting-and-pasting between applications.

Nevertheless, any government or court-ordered solution must confront Microsoft's strong public relations and marketing machine, which means that the product line-based breakup model is at best a steep, uphill battle.

Proposed alternative solution – impose structural changes to Microsoft's business processes, not its organization.

My recommended solution requires looking at the situation from a different perspective – instead of imposing structural changes on the organization, impose structural changes to Microsoft's business processes.

My recommended solution is as follows:

1. Require Microsoft to develop and support versions of its major office products that are fully functional on other popular, current and future operating systems, such as Linux, Java, and Mac OS;
2. Until item (1) is achieved, impose a moratorium on the development and release of the following:
    a. New Microsoft operating systems or significant upgrades to existing operating systems (except for security-related enhancements or upgrades);
    b. Internet Explorer browser (except for security-related enhancements or upgrades);
    c. Office suite product upgrades (except for security-related enhancements or upgrades).
3. Obviously this approach will impose a significant burden on the court or its designated representative to develop and rigorously apply a method for monitoring Microsoft's development activities, both at its own facilities, and at its subcontractor's facilities. Nevertheless, I believe this approach, although not without its challenges, is reasonable and realistic, and, if properly enforced, through a process or mandatory quarterly reporting to the court, is likely to achieve the desired objective.

Some amount of financial profit from the licensing of its products on alternative operating systems is appropriate, as further encouragement for Microsoft's enthusiastic cooperation.

This letter represents a rough outline of my proposal. If you would like to discuss it further, please feel free to contact me. You are free to use my ideas that I have enclosed in this letter in your prosecution of Microsoft's anti-competitive behavior, or in a related matter.

Regards,

Rick Hornbeck

Rick Hornbeck, M.S., J.D.
Hornbeck Consulting
556 S. Fair Oaks Ave., Suite 346
Pasadena, CA 91105
(cell) 323-363-2151

THE TROUBLING TRUTH ABOUT "TRUST" ON THE INTERNET

An objective survey of the security risks associated with ActiveX and its impact on Microsoft's share of the Web browser market.

(by Rick Hornbeck, M.S., J.D. **1997**)*

BACKGROUND

Where did ActiveX Come From and Why Doesn't It Go Away?

By now it is generally accepted that Microsoft and Netscape are engaged in a great World-Wide-Web (WWW) browser war. It is also generally understood that Microsoft's almost limitless revenue from its Windows operating system software and related products will allow it to keep giving it's Internet Explorer browser away free for the next 20 years, while Netscape has to charge customers for it's products. What is less well understood is why ActiveX and the Authenticode securitymodel represent the other two prongs of Microsoft's Internet marketing strategy.

As recently as early 1995 Microsoft was still unsure of the Internet's significance and the role it would play in the PC desktop market. Microsoft believed it could continue its phenomenal year-on-year profit growth relying solely on new sales and paid for upgrades of its existing products. However, these sales must in turn rely on its ability to maintain its grip and influence on the distribution channel, on the corporate purchasers, the original equipment manufacturers (OEMs) and on the standards process. (For example, according to the Microsoft 1996 Annual Report, OEM channel revenues were $1.18 billion in 1994, $1.65 billion in 1995, and $2.50 billion in 1996.) The primary source of OEM revenues is the licensing of desktop operating systems. As such, Microsoft's OEM channel revenues are highly dependent on Windows-compatible PC shipment volume.

During 1995 armies of software developers and consumers launched a blitzkrieg against Microsoft's PC desktop dominion, penetrating the Windows defenses everywhere with dynamically distributed Java applets and gaining over 70% of the market for Internet browsers.

Microsoft quickly realized that the confluence of Java with Netscape's browsers had the makings of a platform-independent de-facto industry standard, which would empower users to buy more non-"Wintel" (Windows operating system on an Intel processor) desktop PCs. The Internet gave Microsoft a vision of it's impending mortality. In response, on Pearl Harbor Day, December 7, 1995, Bill Gates declared war, announcing that "(t)oday, the Internet is the primary driver of the new work we're doing across our entire product line." The Microsoft Web servers took 8,000,000 hits on the first day of their campaign.

After his two-hour public presentation Gates told National Public Radio in an interview, "Well, we've got to make sure that we're leading the way on the opportunities the Internet represents." "Netscape has two great strengths," Gates admitted. "They've got very high browser market share, and they've got the attention of the world. . . . It's very important to increase the popularity of our browser."

Microsoft executive vice president Steve Ballmer put it bluntly when he said, "(h)ave no confusion in your head: Job one for us right now is the Internet and defeating Netscape." Of his Mountain View, Calif., rival in Internet software, Ballmer says, "They're simply our smartest competitor."

It was against this backdrop that Microsoft launched its triumvirate Internet marketing strategy, using the parasitic relationship between Authenticode and ActiveX to increase the popularity of Internet Explorer.

INTRODUCTION

ActiveX and Java are "mini-programs" that can be downloaded from a Web site and executed directly on a user's PC. Unfortunately ActiveX mini-programs, or "components" or "controls" can reformat a user's hard drive, or copy personal files to a remote server on the Internet, or do any number of harmful things to a user's PC without the user's authorization or knowledge. A malicious hacker or terrorist could write one of these downloadable and executable programs and the user-victim has no reasonable way of either stopping it's attack once the control has gained access to their PC or reliably preventing it from gaining access in the first place.

The user has several "unreasonable" means of minimizing her risk: she can permanently disconnect her PC from the Internet, depriving herself of its benefits. She could browse only those Web sites that she "knows" do not contain harmful or malicious controls ("safe zones"), although the possibility of a hacker either spoofing a Web site, or covertly placing harmful controls into a "known" Web site exists. She could configure her Internet Explorer browser to prevent all ActiveX controls from downloading to her PC, and hope she does not encounter one that is able to bypass her browser's security configuration, which has been demonstrated in practice. Finally, she could take her chances using Microsoft's "Authenticode," or Netscape's or Sun Microsystems' "code-signing", trust-based security models that use public-key digital signatures and independent third-party Certification Authorities (CAs). Each of these unreasonable alternatives represents a different point on the risk/benefit scale which each user should consider before exploring the WWW.

However, this analysis is only necessary because Microsoft created a previously non-existent security risk by introducing ActiveX. As will be explained below, other software tools exist to provide software developers with the same capabilities as ActiveX, with virtually no security risk. Still, Microsoft has successfully obfuscated the seriousness of these self-created security issues and successfully redirected consumers' attention away from Netscape and Java. In doing so Microsoft has also successfully achieved its goal of creating the perception, in a very short period of time, that it is a player in the Internet game.

Because ActiveX does not contain its own internal security mechanism to restrict the actions of the program, Microsoft was able to introduce the Authenticode trust model as a viable protection solution. Because Authenticode uses public-key digital signatures in combination with trusted third-party Certification Authorities, and only runs on Internet Explorer, Microsoft sought to "increase the popularity of its browser" by touting its use of this "cutting-edge" technology as evidence of its leadership in the Internet software industry. At the same time it actively castigated Netscape and other browser vendors for allegedly leaving their users vulnerable to the hazards of ActiveX. Unfortunately, the people that suffer from this Machiavellian marketing strategy the most are the innocent netizens who "reasonably" allow unproved and potentially dangerous controls to be downloaded to their PCs leaving themselves vulnerable to the vagaries of malicious programmers.

It would be too harsh to accuse Bill Gates of raising Microsoft to is position of dominance through villainy or malice against his customers, given the trends of modern business practices. However, his continued promotion of Authenticode without acknowledging its serious security defects would seem to indicate that its effectiveness in mitigating security risks is subordinated to creating the impression that Microsoft is a leader in the Internet/Electronic Commerce industry. According to Eric Schmidt, Novell CEO, "if Bill Gates continues with his strategies he could become the most powerful person in the world, and that's not necessarily a good thing." Simson Garfinkel wrote recently, "Microsoft's ActiveX technology is the single greatest technological threat to the future of the World Wide Web. Microsoft's ActiveX promoters are either so blinded by their own rhetoric that they don't see the danger of this new technology, or else they are so cynical that they would destroy the very essence of the Internet rather than compromise their market dominance."

In a different industry, Microsoft's actions could be analogous to a pharmaceutical/bio-engineering company releasing a virus or disease into the general population so it could profit from the sale of its potential cure. At the same time the pharmaceutical company could also enhance its reputation by advertising that it's anti-virus was created through the use of cutting-edge genetic engineering techniques thereby establishing itself as a leader in this field. However, for this analogy to be consistent the anti-virus must only be effective for a small percentage of the population. The rest of those exposed to the virus would remain susceptible to its deadly effects at any time.

This article will explore the very real damage that can be caused by harmful ActiveX controls. It will explain how Authenticode is supposed to mitigate these security risks, and why it does not. It will also explain why digital signature technology as currently applied under the Authenticode model cannot assist most users in adequately reducing their risk of injury from ActiveX because it does not provide the user with the necessary means of assessing whether or not the software they are considering downloading is "safe."

Bill Gates' vision of the future is a seamless integration of the Internet, the World-Wide-Web and the Windows operating system. According to Gates, when someone wants to e-mail a spreadsheet or other file to someone else over the Internet, they are not interested in going out and buying 14 different products to make sure the file will be compatible with the recipient's software. Instead what they want is a desktop environment that can provide spreadsheet and any other kind of robust functionality, without concern for the software or hardware on which it operates.

Most Internet software developers' share this vision however they don't share Gates' vision for implementing it. Microsoft believes this seamless integration should be based on Windows and Microsoft's Internet Explorer (IE) browser whereas the rest of the software industry favors Java because of its true platform-independence. Today Java can run on virtually any hardware or software platform in existence, including such varied platforms as IBM mainframes and Personal Digital Assistants (PDA's).

Yet Microsoft continues increasing the popularity of its proprietary browser by:

- Marketing the benefits of ActiveX while simultaneously cross-marketing Internet Explorer (IE) because IE is the only platform capable of directly running ActiveX controls;
- Continuing to give its IE browser away for free;
- Failing to live up to its promises made in the fall of 1996, to disclose ActiveX's specifications to an independent standards body, thereby preventing other browser manufacturer's from supporting it in their products;
- Marketing IE as the only means available for user's to purportedly protect themselves from the potential damage threatened by its own hazard, ActiveX, and
- Cross-marketing Authenticode as a general-purpose Internet security solution, thereby further reinforcing the perceived need for IE, because it is the only browser capable of supporting Authenticode.

A BRIEF COMPARISON AND CONTRAST BETWEEN ActiveX, JAVA AND PLUGINS.

(1) Origin of ActiveX

ActiveX adds to the user's Internet Explorer-based Web browsing experience by "jump-starting" Web site content, providing a variety of multimedia effects, enhanced page layouts, and executable applications, all of which are downloaded and run in real-time over the Internet. According to Microsoft, over 1,000 ActiveX controls already have been written in C, C++ and other languages for applications such as audio, video and live chat, all of which complement the core technologies of today's Web environment such as HTML, plugins, Java, cgi scripts and more.

According to Fred Langa, writing in Windows Magazine, ActiveX is ". . . the fifth and most recent step in a long-developing evolution (by Microsoft Laboratories) of data-sharing and interoperability among applications." Essentially it is a trimmed down version of Microsoft's OLE (Object Linking and Embedding) system which a Windows "power" user will recall enables several applications to collaborate on a single "compound document." For example, OLE provides the "glue" that allows data to be copied from a WORD document and pasted into a PowerPoint document. The PowerPoint document can then be inserted into an Excel document and later opened as a PowerPoint document from within the Excel document. ActiveX is the next step in the development of this seamless interaction amongst applications. However, where "Distributed OLE" only lets the user share data, links and control over a local or wide-are network, ActiveX has taken the technological "leap" into Cyberspace by enabling the user to share data, application links and control between a Web page on the Internet and the user's Internet Explorer browser running on his PC. Java has taken the same leap but with much less risk to the user.

ActiveX controls automatically download and install themselves, and they persist (remain available) on a user's system. This feature provides two advantages over other programs: the user doesn't have to download and install software manually, and she only has to download the control once. This is good news to those who don't like waiting for controls to download every time they visit a certain site. However, these controls can be downloaded without user awareness or consent which means the user doesn't know what she is downloading.

(2) JAVA

Java applets can be thought of in the same way but with some important differences. Java applets run either inside the Java Virtual Machine (JVM), a software application that is built into newer browsers, or they can be run separately using the Java Development Kit (JDK). The JDK is a sort of software interpreter that converts Java code into code that is recognizable by the particular platform on which it is running. JDKs are now available for virtually all software and hardware platforms in existence. However, because JDK is another layer of software between Java and the actual operating system, Java tends to run more slowly. "The major fear is that Java is not going to have the performance it promises, and its going to fade away like a bad TV show."

Built into both the JVM and the JDK is a set of security controls colloquially called the "sandbox." Java's security model automatically prevents any code from accessing portions of the operating system or the PC hardware that is outside the parameters of the "sandbox." In other words if a Java applet wants to "play" on your PC it has to keep its toys inside the sandbox. In contrast, ActiveX controls are not restricted, which means they have direct access to the PC hardware, software and operating system. As a result, ActiveX controls run faster and do more, but at a substantial price in security. Also, because ActiveX controls are distributed in native binary code, separate controls have to be written for each operating system. Java applets, on the other hand are distributed in a one-size-fits-all or "write once, run anywhere" fashion meaning that developers only have to produce one version to run on any platform.

(3) Plug-ins

A third means of "activating" a Web site is through the use of Netscape "plugins." Both Netscape and Internet Explorer browsers are packaged from the factory with a built-in set of "standard" features such as graphics viewers, which a Web site developer can then take advantage by including graphics in his Web site. However, in order for a Netscape browser to take advantage of any non-standard features which the Web site developer has programmed into his Web site, the "plugin" version of the entire application that is used to run it must first be downloaded to the user's PC from the developer's Web site and then executed. This is because the application is not embedded with the program, as in the case of ActiveX.

For example, assume that both an ActiveX control and a non-ActiveX program using plugin technology are created to enable users to download and view a short animation sequence from a commercial Web site. The ActiveX developer will include both the animation sequence and the "viewer" program in the same control. However, the developer using plugin technology must create a built-in hyperlink in the code to the viewer developer's Web site. When the user clicks on the link on the Web site to view the animation sequence, the code will automatically notify the user that she must go to the vendor's Web site and manually download the entire "viewer" software application before she can see the animation. ActiveX components are inherently much smaller because they contain only a limited subset of the entire application needed to perform the function at hand, and therefore can be downloaded more quickly. Once the ActiveX component is resident on the user's PC it can be reused, on-demand precluding future downloads.

According to Microsoft, the excessive amount of time needed by a user to download the actual application "plugin" file (.exe) poses a significant deterrent to the use of

Netscape's browsers. However, as described in an article in the May 27, 1997 issue of Fortune magazine, Netscape's new Communicator browsers will also automatically install "plugins."

ACTIVEX's SECURITY DEFECTS ARE "GENETICALLY INHERITED"

Because ActiveX is the product of many years of ongoing research and development at Microsoft laboratories it represents the latest in a long line of remarkable software technologies. However its predecessors, OLE and COM, have burdened ActiveX with their "genetic blueprint," legacy code written for earlier generations of software and hardware platforms. In other words this latest progeny is constrained by its "gene pool" consisting of thousands of lines of code which have accumulated over the course of years of development and evolution and over which ActiveX is unable to break free. The most significant constraint imposed on ActiveX by this genetic blueprint is a deficient security model. According to Microsoft:

> We are doing everything possible to create the technical safeguards that will make software safe. However, in order to remove trust from the equation, we would have to rip away significant amounts of functionality (read: code that could actually be rewritten to more closely fit the needs of the modern Internet environment) that users rely on today. Since the purpose of our industry is to provide more value and power to users, rather than limiting functionality, Microsoft and most other major software manufacturers are advocating a trust-based security model. (read: we could rewrite it if we wanted to but because it doesn't suit our interests we won't)

This "genetic" deficiency allows ActiveX controls to interact without constraint with both the operating system and the PC hardware. In a sense, it is as if ActiveX was born without an auto-immune system, making it incapable of combating viruses or malicious programming written by evil programmers that might invade the control and use it to enter and harm an innocent, unsuspecting host.

ActiveX's predecessors did not have to be concerned with such an auto-immune system because they were virtually guaranteed of living out their lives in a "sterile" environment. In other words, prior to the advent of the Internet the operating environment in which PC software was executed was always under the complete and exclusive control of the PC user. Each user was able to decide whether they wanted to load a particular program on to their PC, and once loaded whether and when to execute it. This environment remained "sterile" regardless of whether or not the PC was a standalone or networked because no external source, including a network operating system could place something onto the users PC without his or his network administrator's permission.

Today, however, through the wonders of downloadable and executable software technologies, a program can automatically download to a user's PC from a Web site or a network server and execute without the user's awareness or consent. Thus, the operating environment in which Microsoft's next generation software tool is living is completely different than the environment of its forefathers. Yet Microsoft has chosen not to take this congenital auto-immune deficiency seriously and has failed to reengineer ActiveX's "DNA" to create a reasonable security model thus leaving users vulnerable to exposure to the dangerous code. Such an unprotected and infected control acts like a cyber "Typhoid Mary" as it infects everyone it meets with the virus of harmful code.

By way of explanation, suppose Mr. And Mrs. Jones owned and lived in a house during the same time the Microsoft software engineers were developing the ancestors of ActiveX. Mr. Jones worked diligently on his house, making improvements and refinements so it would be more comfortable for he and his wife. Now suppose Mr. And Mrs. Jones decide they want to start a family and Mr. Jones asks a contractor for a cost estimate to build a second-story bedroom. The contractor tells the Jones' that because their house was built using an "A frame" design a second story cannot be added. Thus, the Jones' are constrained from meeting their needs for another bedroom by the limitations of their house's original design, which did not take into consideration the future need for a second story. Similarly, ActiveX is constrained from incorporating a security model bythe limitations imposed on it by the software designs of its predecessors.

However, if Bill Gates were the owner of this "A frame" and he wanted to add a second story because he and his wife wanted to start a family, he could easily afford to tear down the existing structure and build whatever design fits his current needs. Similarly, Mr. Gates and Microsoft have the resources to re-write ActiveX or develop a replacement. Indeed, one can only speculate why he has chosen not to develop an Internet software product that fits the current needs of his customers, given that the environment in which his software executes (the Internet) has changed, and is now "open" and "insecure."

Without providing an answer to this rhetorical question, Jesse Berst also observes in PC Week, "ActiveX is . . the key to its future. Microsoft will be damned before it acknowledges that ActiveX has a security problem." Berst goes on to explain that "(r)ather than help users understand and minimize the risks (associated with ActiveX), Microsoft contented itself with pointing out that similar problems were theoretically possible with Netscape products." Quoting PC Week Editorial Director and former director of PC Week Labs, David Berlind, Berst writes, "Frankly, I want to puke."

Microsoft will not give up ActiveX because it is the key to "increas(ing) the popularity of its browser." Without ActiveX there would be no need for Authenticode, and without Authenticode and ActiveX there would be no way of significantly distinguishing IE from a Netscape browser, except that it is given away at no immediate up-front cost.

THE AUTHENTICODE SOLUTION - Myth and Reality

(1) The Myth

In his article Jesse Berst explains that Authenticode is ". . . like requiring people who send mail bombs to put their names on the package." Were that approach effective, even the alleged "Unabomber" would have been apprehended many years earlier, because according to news reports many of his mail bombs had postmarks from the small town where he lived. Obviously this approach is ineffective because the names would be blown up, just as any evidence of an Authenticode digital certificate could also be destroyed by a malicious ActiveX program after causing other damage to a user's PC. And yet on August 7, 1996 a Verisign Press Release quoted Verisign president and CEO Stratton Sclavos as stating, "With this service, users can feel confident that the applications they receive are authentic and tamper-proof." The same Press Release also quotes Sclavos as stating that, "Under the Authenticode program, developers must go through an application and verification process to ensure that certificates are issued only to the appropriate party. This eliminates any worry that developers could be falsely represented by an impostor."

Microsoft's Authenticode security model requires that all software developers (commercial and independent) must register their ActiveX components with a Certification Authority such as Verisign, before Internet Explorer browsers will allow them to be downloaded to a user's PC from a Web site, if the browser's security setting is on 'High.' The software developer must "legally" affirm that to the best of his knowledge the control is incapable of causing damage to a user's PC. Verisign issues the developer either an electronic "Individual Software Publisher's Certificate" or an electronic "Commercial Software Publisher's Certificate" depending on whether they are registering as an individual or corporate software developer. Different identity verification criteria are used to establish the developer's identity depending on the type of certificate requested.

One way Microsoft successfully propagates the Authenticode myth is through contradictory and vague announcements and bulletins. The following excerpts demonstrate the range of conflicting statements about Authenticode that come from both Verisign and Microsoft management.

The following excerpt from Verisign's Web site explains the service it provides to its customers:

> When customers buy software in a store, the source of that software is obvious. Customers can tell who published the software, and they can see whether the package has been opened. These factors, along with others, enable customers to make judgments about what software to purchase and use, and how much to "trust" those products and the companies and individuals who publish them.
>
> When customers download software from the Internet, all they see (at most) is a message warning them about the dangers of using the software. The Internet lacks the subtle information provided by packaging, shelf space, shrink wrap, and the like. Without an assurance of the software's integrity, and without knowing who published the software, it's difficult for customers to know how much to trust software. It's difficult to make the choice of downloading the software from the Internet.
>
> Verisign Digital IDs in conjunction with Authenticode (software validation) technology provide customers with the information and assurance they need when downloading software from the Internet. Authenticode communicates to customers the real identity of the publisher and assures them that the product has not been altered or damaged. (emphasis added)

Contrast this language with the statement of Cornelius Willis, Microsoft's group product manager-Internet developer marketing, "Authenticode does not guarantee that users will never download malicious code to their PC. . . . We don't claim ActiveX is a completely safe environment. If (a control) can get at your hard drive it is not totally secure."

(a) The Problems of Establishing Identity in Cyberspace

The advantage of knowing the publisher's true identity is that it provides the relying party with recourse in the event the software turns out to be "harmful." In the physical world this is generally not a problem, as a purchaser can usually assume that the store's physical location will not change. The benefit of having a physical location to return to serves several purposes. First, the store owner's physical assets can be attached; second, the unsatisfied consumer can create a scene inside the store, or in the community, creating bad publicity for the owner and an incentive for prompt resolution; third, the physical location will be an indicator of the laws that will apply in the particular jurisdiction.

Transacting in a physical location has advantages for the seller as well. The merchant can demand physical identification which can usually be verified through on-line databases combined with visual scrutiny of a photo ID, the purchaser's demeanor and dress and other non-verbal cues which can be stored by a video camera for future retrieval and proof of the transaction should the purchaser later attempt to repudiate.

Telephone-based sales represents a hybrid marketplace with portions of the physical world and Cyberspace. From the consumer's standpoint, if she dialed an 800 or 888 number she has little assurance of who she was actually calling, where they are located what laws apply, and whether the "order taker" works for the company she is purchasing the product from, or an outsourced tele-marketing firm. The risks to the consumer are only that she may be giving her credit card number to someone other than a legitimate merchant who will use it fraudulently. However, her exposure is minimal because most credit card companies limit the consumer's liability to $50, assuming timely, good faith reporting efforts.

The merchant suffers greater risks through telephone-based sales, although the tradeoff is less overhead than a storefront. If the consumer dials an 800 or 888 number, 'caller id' will notify the merchant of the phone number used by the purchaser to make the call which can be used in connection with reverse phone directories and address cross-checking databases to provide additional identity verification. However, the merchant is unable to demand visual identification, and is legally protected only by on-line credit card clearing services, which can only benefit the merchant after the credit card theft has been discovered and reported. The majority of credit card thieves use the card as quickly as possible after the theft to take advantage of delays in reporting. Because of the limitations on identify verification, and the delays in theft reporting, the likelihood of fraudulent telephone-based transactions increases significantly.

Internet-based sales represent the greatest opportunity for fraud to both parties. The merchant is unable to establish the caller's telephone number and related identifying information. Telephone records cannot provide evidence that the phone call took place because access will be through an independent Internet Service Provider dial-up service. Although Web servers can gather user information through cookies this is not always reliable. The opportunity for using stolen credit cards is at least the same as with telephone sales. (See "The Essential Role of Trusted Third Parties in Electronic Commerce," Michael Froomkin)

Also, it is possible for a Web site to be "spoofed" or misrepresented by a hacker, causing the unsuspecting user to enter their credit card and other relevant identifying information on-line. Although a technical discussion of "Web spoofing" is beyond the scope of this article, a "spoofed" Web site can look exactly like the original to anyone but the most cautious of users. The unsuspecting consumer personal data would be turned over to the thief who would quickly use it.

Because of these and similar identity authentication problems consumers and merchants cannot know with certainty, even with Digital Ids, the actual identity of someone on the Internet. Greater security measures are needed before consumers can reasonably trust the Internet as a medium for safe commerce.

AUTHENTICODE - THE REALITY

WHAT IS THE ROLE OF THE CERTIFICATION AUTHORITY?

The purpose of a Certificate Authority is to bind a public key to the common name of the certificate, and thus assure third parties that some measure of care was taken to ensure that this binding is valid. A measure of a Certificate Authority is their "Policy Statement" which states what measures they take for each class of certificate they offer to ensure that this binding of identity with public key is valid.

2. WHAT IS THE ROLE OF A DIGITAL ID (PUBLIC KEY CERTIFICATE)?

Although the actual digital signature process will not be covered in detail, the following brief explanation will highlight some of the important points. Traditional encryption for confidentiality uses only a single, 'secret' key and is called symmetric cryptography. Digital signatures use a mathematically related key pair, (a 'public' key and a 'private' key) and employ a technology called asymmetrical cryptography. A mathematical formula or algorithm is used in conjunction with a 'random-number' generator to create the public and private keys. The design of the encryption algorithm relates the two keys in such a way as to allow either key to decrypt a message encrypted by the other. However, it is 'computationally infeasible' to determine the value of the private key based on the public key and the digitally signed message. Additional information on digital signature is available at www.rsa.com and www.abanet.org/scitech/ec/isc.

The utility of a digital signature as an authenticating tool is limited by the ability of the recipient to ensure the authenticity of the key used to verify the signature. The following explanation will demonstrate this truth. The traditional labels used to represent the different parties in this sort of discussion are Bob, the sender, and Alice, the recipient. For purposes of this discussion a third party, Mallet, will play the role of evil hacker.

If Bob digitally signs a message using his private key and sends it to Alice the only way she has to verify that Bob really sent it is if she knows Bob's public key. However, Alice must be able to retrieve Bob's public key from a source other than Bob's message because if Mallet is forging Bob's message he will send his own public key, claiming that it actually belongs to Bob.

Mallet has the private key corresponding to the public key sent to Alice, her attempt to authenticate the message will result in a positive confirmation even though it was not really from Bob. However, if Alice has access to Bob's real public key from an outside trusted third-party source, and uses it to verify the message signed with Mallet's private key, the verification will fail, revealing the forgery. In short, the Certification Authority (CA) fills the role of an outside source and Bob's public key is transmitted from the CA to Alice in the form of a Digital ID or public-key certificate. In order to ensure the authenticity of the certificate, Bob's Digital ID will be digitally signed by the CA. In order for Alice to establish a "trusted" relationship with the CA she must have access to the CA's public-key from another trusted third-party.

In practice, most if not all CAs have chosen to provide their public-key certificates to Netscape or other browser developers, who embed them into their browsers for easy access. In the event Bob has registered his public-key with a new, or unregistered CA, the browser software will notify the user and give him the opportunity to accept the CAs public-key 'on the spot.' This presents the user with a predicament, and also presents CAs with a strong incentive to pre-register with the Netscape, IE and other browsers.

The fundamental problem comes down to how good a job the CA did in authenticating the subscriber identity. The CA's response will be that it made a good-faith effort consistent with the terms of the agreement or CPS to which both parties are bound. However, close scrutiny of the agreement will reveal that (1) very little detail is provided about the authentication methods used or the reliability of its sources of information, (2) the level of effort invested in the identity verification process is a function of the Level or Class of Digital Id. In other words, a subscriber's Digital Id that costs $20 will not receive as much identity authentication effort as will the subscriber to a $400 Digital Id. The following examples are cited by Verisign as representative of the sorts of transactions that could reasonably be performed using the various Levels of Certificate:

These examples, as well as any attempt to standardize on a generalized template of reasonable reliance is of marginal utility. It quickly breaks down when faced with simple counter-examples such as the following. According to the Verisign Digital Id Certificate model, a Class 1 Digital Id is acceptable for use in confirming the identity of e-mail correspondents and transactions of very low value. Assuming an organization chose to use the Class 1 Id for transactions that are limited to a value of $.01, but the number of these transactions exceeds one million per day. Under these facts the company

3. HOW DOES THE INTERNET EXPLORER BROWSER PROCESS THE DIGITAL ID?

The following step-by-step explanation of what happens when an Internet Explorer browser visits a Web site containing an ActiveX component will provide an overview of the basic steps involved in the public-key digital signature process, as applied in Microsoft's Authenticode model. Additional introductory material on the subject is widely available on the WWW, including the Verisign, RSA, and American Bar Association, Information Security Committee sites.

> When the IE browser arrives at a Web site that contains an ActiveX control the browser will first check to see if the component has been digitally signed.

If not, the browser will display a warning message to the user, stating that the component is of unknown origin and may present a security risk, and then allow the user to make the choice whether to allow the component to be downloaded to their PC or not.

If the component has been digitally signed the browser will determine which Certification Authority (CA) authenticated the certificate, and if it doesn't already have a stored copy, it will automatically obtain the software publisher's public key from that CA via the Internet.

The browser will then use the public key to decrypt the "message digest" portion of the certificate.

The browser will then run the same digital signature "hashing algorithm" on the component again and match the resulting message digest against the one in the certificate.

If the component has not been modified, either intentionally or inadvertently since it was signed, the new digest should match the old one.

If they don't match, either the code was modified or the public and private keys aren't a matched pair. Either way, the component becomes suspect and the browser notifies the user that it should be discarded.

4. PROCESS WHEREBY SUBSCRIBER CONTRACTS WITH A CERTIFICATION AUTHORITY FOR A DIGITAL ID.

The subscriber must provide the Certification Authority with enough identifying information to satisfy the CA's authentication requirements, depending on the Certificate Class. For example, the following information must be provided to Verisign during the enrollment process, either through their on-line enrollment forms or through regular mail.

Individual Software Publishers (Class 2):

- Individual Publisher's name, address, and e-mail address
- Date of birth
- Social Security Number
- Previous address (if you have moved in the past 2 years)
- Credit card information for billing

Commercial Software Publishers (Class 3):

- Company name, address, e-mail, phone, and fax
- information for a technical contact and an
- organizational contact.
- company's DUNS number, if any.

- *Billing information (credit card, P.O. or check), and billing contact* information, if any.

As of June 1997, pricing for Software Publisher Digital IDs are as follows. Digital Ids for different purposes are also available, at different prices.

Class 2 Digital ID for Validating Software: $20 annually (for Individual Software Publishers)

Class 3 Digital ID for Validating Software: $400 annually (for Commercial Software Publishers, i.e. companies)

The following excerpt from the Verisign Web site explains their procedure for verifying a company or individual identity.

> Based on Microsoft code signing program criteria, VeriSign will attempt to verify that your company meets a minimum financial stability level using ratings from Dun & Bradstreet Financial Services, or attempt to verify your personal information through a credit reporting agency such as Equifax for individual software publishers. Your certificate will indicate if you have met this level. Some software, such as the Microsoft Internet Explorer 3.0, offers end users an option to bypass making an explicit choice to trust code from each new software publisher. If an end user checks an option to trust all software signed by vendors who have met the financial criteria, code signed by these vendors will be run without any user intervention.

5. THE UTILITY OF AN AUTHENTICODE DIGITAL ID

All properly authenticated digital signatures can demonstrate to a high degree of certainty the following three attributes:

Integrity - The component has not be modified since it was signed, either intentionally or inadvertently.

Authentication - The purported identity of the party who registered as the component's author, based on the certificate's level of assurance and Verisign's corresponding identity verification criteria.

Non-repudiation - The component's registered author cannot later repudiate his identity as the component's registered author should it cause damage to a user's PC or other computer-related product (assuming the author registered the component using his own identity).

However, because Authenticode will only work on Microsoft's Internet Explorer, users of any other browser will be unable to gain whatever benefit might be provided by this information. For example, if the ActiveX plugin from NCompass Labs, Inc. is used with a Netscape browser, any ActiveX component encountered on a Web site by the browser will be downloaded without Authenticode's intervention. Netscape's generic software download alarm will probably display a warning, giving the user an option to proceed or quit, but the existence of a Digital ID will not be a factor in the user's decision.

Digital Certificates can only attempt to vouch for the authenticity of someone's identity, not for their good intentions. Neither the digital signature technology nor the Certification Authority (CA) make any warranties as to the safety of the ActiveX component. The Authenticode system merely relies on the assurances made by the component's developer to the CA when they initially apply for a Digital ID subscription. In the patois of logic this appears to be circular reasoning. The party whose trustworthiness is in question is providing the means for assuring the user of his trustworthiness. Furthermore, CA's have neither the mandate, resources, nor the incentive to actively monitor the behavior of millions of its certificate holders. Although they do have a duty to suspend or revoke a subscriber's Digital ID based on reported breaches of a specific set of criteria, they are not obligated to perform an independent monitoring function.

The possibility of undiscovered fraud is significant due to the ubiquity of stolen credit cards and access to personal information on the Internet combined with the limited authentication of the user's identifying information. Authenticode is supposed to provide the means for a user or corporation to "trust" the ActiveX components they download from the Internet by ensuring "accountability."

> The approach here is <u>accountability</u> -- to cease having publication of software on the Internet be an anonymous activity. If an organization or individual wants to use the public Internet to publish software, they <u>should</u> be willing to take public responsibility for the code they author and publish. If the code proves to have errors or even malicious faults, these organizations and individuals should be willing to answer for them just as they would take credit for good code. This approach is founded on the idea that accountability is an effective deterrent to the distribution of harmful code. (emphasis added)

The same argument can be made that license plates should act as deterrents to either prevent or curtail the use of cars in the commission of crimes. Because the license plate establishes the owner's identity (with possibly more certainty than a software publisher's certificate) it makes him accountable for his acts using the car and therefore cars will not be used in the commission of crimes. Still, <u>stolen</u> cars are used every day, to smuggle drugs, transport criminals to and from crime scenes, and perform other illegal acts. Obviously accountability is not an effective deterrent to the use of cars to commit crimes. Likewise, accountability is not an effective deterrent against the malicious use of ActiveX, because stolen credit cards are readily available.

What is the solution to this problem? There is probably no single solution short of eliminating ActiveX entirely. However, a number of individual solutions are appearing which, when used in aggregate have the potential to reduce the threat of injury to an acceptable level. Several of these potential solutions are discussed below.

6. DIGITAL AUTHENTICATION FOR WEB SERVERS.

Verisign, Xcert, GTE and other companies are also in the business of selling Digital Ids for Servers. According to Verisign, their product would enable the server owner to establish his authenticity to Web browsers visiting his site. In the marketing literature describing Digital Ids for Servers on its Web site, Verisign explains:

> In the virtual world of the Internet, however, the web-site of an unscrupulous con-artist might look just as professional as that of a legitimate business. The low cost-of-entry and the ease with which graphics and text can be copied make it possible for almost anyone to create sites that appear to represent established businesses or organizations. To protect your organization and your customers from such impostors, you need a way to establish you site's authenticity.

Interestingly, in one context Microsoft and Verisign guarantee that users will be able to garner enough information by visiting the developer's Web site to make an informed judgment of both the developer's and his program's trustworthiness. However, in this context Verisign is saying that because almost anyone can create Web sites that appear to represent established businesses or organization that Web site owners should use Digital Id for Servers to establish their site's authenticity to visitors.

Later in this same Microsoft document mentioned above, under "Qualifying for the Individual Software Publishing Certificate" Microsoft rhetorically asks the question, "What is the value of the Individual Software Publishing Certificate?" The document responds:

> It would seem that users aren't going to trust individuals they don't know, and businesses aren't going to let code signed by students at a local university into their corporate domain. While this may indeed be the case, the value of this type of certificate is in the information it provides to the user so that he/she can make the decision on how to run the code. Knowing who authored the code, and that the bits have not been altered from the time the code was signed to the present is indeed comforting information. Additionally, the implementation provides links from the user interface (UI) to Web pages so the user can obtain detailed information about the signed code, the author, and the certificate authority. After learning about this code and the author, the user may decide to run the code, and/or all future code signed by this certified individual. (emphasis added).

Leaving aside the remarkable statement that corporations would inevitably not allow software developed by local university students into their domain, Authenticode fails to provide an objective means for users to evaluate this supposedly detailed information about the signed code and its author that is being made available to them. One is left with the gnawing suspicion that Microsoft intends for there to be a direct relationship between a software developer's advertising budget, the purported "trustworthiness" of his software, and the frequency with which users will download it over the Internet. In other words the more a developer can achieve brand name and product name recognition amongst Internet users the more frequently his products will be downloaded. Not surprisingly, Microsoft has one of the biggest advertising budgets in the world.

7. PULLING IT ALL TOGETHER WITH SSL, . . . ALMOST.

We have seen that browsers can authenticate software publisher Digital Ids and that Web servers can authenticate client browser Digital Ids, assuming the subscriber's identity is established with reasonable certainty. However, this authentication is only performed once, at the beginning of the transaction. After the initial "handshaking" takes place and the browser software is convinced that the other party is who she claims to be, no further checking is performed. This would leave either or both parties vulnerable to eavesdropping, replay and spoofing attacks during the remainder of the communication, if not for SSL.

Secure Sockets Layer (SSL) is an industry standard communications protocol that attempts to remedy these problems by creating unique signature keys that are exchanged throughout the entire communication "session." In other words, after the client is certain the server is not spoofing its identity, the server and client exchange "session-keys" that will be used to sign the data during the data exchange. With SSL 2.0, the same signature keys must also be used for encryption, if confidentiality is needed, however with SSL 3.0 signatures can use different keys than the encryption engine.

SSL's main function is to protect users from attack by eavesdroppers or message interceptors. Both the client and the server provide part of the random data used to generate the keys for each connection and that same random data is also used to generate the master secret key associated with that session.

(a) Caching data during secure connections

One important drawback to this SSL scheme is the fact that the Netscape browser can store in local cache on the user's hard disk any data that has been sent by it during the secure connection. Navigator 3.0 has an option to allow caching of data fetched over SSL connections, however the default setting is to not cache data. In Navigator 2.0, documents fetched using SSL were cached in the same way as non-SSL documents. However, the command "Pragma: no-cache" in the HTTP header can be used to disable caching for a particular page. Interestingly, in Navigator 1.0 documents fetched with SSL were not cached.

Most importantly the cached data is not encrypted and is available to "prying eyes" in cleartext form. As long as the cache remains on the user's hard disk, any information such as credit card numbers or private keys that were sent over the secured SSL connection are ripe for the picking by anyone either physically accessing the PC or using an intermediate agent such as an ActiveX control.

(b) Handling previously unknown certification authorities while Web browsing

Whenever a previously unknown CA is encountered by a browser their Root keys for Certificate Authority certificates are loaded through an automatic process using an SSL connection. This means that conceivably a 'rogue' CA can load its certificate into browsers and begin authenticating harmful ActiveX controls without any restrictions. Netscape states that presumably in the future loading a root certificate through a local process, such as from disk, LDAP, or other out-of-band mechanism, will be a supported addition or in place of the present method of connecting to a trusted server and downloading the certificate chain. This presumption is an acknowledgment of the severe security risks associated with the current approach, and also an acknowledgment of the technological complexity of the more secure approach.

(c) Vendor Incompatibles

The successful application of these SSL keying standards is also completely dependent on the capabilities of both the client browser and the Web server. However because different software vendor's products support different implementations and versions of

SSL, fundamental barriers still exist to prevent a universally "secure" Web browsing experience. Other obstacles to trustworthy applications include the inability for Web servers to automatically check every certificate for currency, either by checking its expiration date, or checking an on-line "certificate revocation list" (CRL) to determine whether the certificate has been suspended or revoked for fraudulent or criminal abuse. As this technology evolves, these barriers will be eliminated, bringing us closer to the goal of authenticated, safe communication on the Internet. The problem in the near term however, is that most users are not made aware of the risks associated with these technological shortfalls.

8. CERTIFICATION REVOCATION LISTS (CRLs)

A certificate revocation list (CRL) is a repository of information that presents the current state of any public-key certificate to anyone who accesses it. The CRL can be implemented in different ways but the approach Verisign uses for the Authenticode Digital Ids is to only include those certificates that have a current unrevoked status. In other words, it is possible for a certificate to either be in an active, suspended or revoked state. If the certificate has been revoked it should not be relied on under any circumstances. However, if the certificate is temporarily suspended it is possible that removal of that status is imminent and the potential relying party should contact the Certification Authority directly for further details. Regardless of the unique circumstances it is essential the potential relying party have access to the certificate status or he will be making an uninformed decision regarding reliance. Implementation of the CRL is another contentious subject that again trades off between the development costs to provide customer ease-of-use and informed decision making. Unless the potential relying party knows how to access and use the CRL they are unable to benefit from its contents. However, instructions on its location and use are not conspicuously displayed when the potential relying party is presented with the publisher's Authenticode-based Digital Id. This is generally because this option has only recently been made available to HTML programmers and so a significant retrofitting of all certificates is needed to implement it.

When implemented properly a button will appear on the Document Info page for servers whose certificate supports the appropriate extensions or commands. When the button is pressed the CA will be queried via HTTP GET, and will display a dialog to indicate to the user if the certificate is good or not. This button does not appear in the Authenticode Digital Id but instead must be "manually" selected from the "View" pull-down menu on the browser. If a user attempts to use a client certificate that has expired, a dialog will be displayed warning them that their certificate has expired, and if this extension exists, a button will be on the dialog that will bring up a window displaying the URL.

There is no automatic revocation check. As mentioned above, a button allowing manual checks is displayed on the Document Info page. According to Netscape this feature was added because some people needed revocation, but they did not have time to support full CRLs. However, in a future release they will support CRLs, and possibly other forms of revocation technology.

Client authentication as implemented by Microsoft Internet Explorer 3.0 is interoperable with popular Web servers that support secure sockets layer (SSL) 3.0 client authentication. Microsoft is working to extend the complete set of technology components necessary for webmasters to incorporate client authentication in their Web applications. This includes extending Windows NT(r) Server operating system support for challenge and response and the SSL 2.0 protocol used by Microsoft Internet Information Server to also include support for client authentication through the SSL 3.0 protocol.

7. RELYING PARTY AGREEMENT

The greatest potential victim of any defects in the Authenticode model is arguably the relying party who attempts to verify the Digital ID and make the decision to download. A detailed discussion of the many legal uncertainties surrounding CAs and certificates is beyond the scope of this article. Suffice it to say that a legal outcome will in part depend on the jurisdiction hearing the claim and the "reasonableness" of the reliance. Verisign has attempted to address many of these issues in its "Relying Party Agreement" which, according to its language, is binding as soon as the third party "relies," either intentionally or otherwise. This reliance is supposed to be triggered automatically when the party inspects a Verisign Certificate Revocation List or accepts a Verisign Digital ID. This agreement also attempts to remove the "choice of law" or jurisdiction question by specifying that all parties are bound by California laws. However, a more fundamental question must first be addressed. Under California's Uniform Commercial Code (UCC) statutes however, if a certificate is considered a good rather than a service, any disclaimer of warranties must consist of a conspicuous writing attached to the good being sold. It is difficult to envision how this should be accomplished, yet Verisign's incorporation by reference may not meet the California standard for conspicuousness.

Furthermore, the relying party is expected to read this agreement before "us(ing) or rely(ing) upon any information or services provided by VeriSign's Repository or website" or "search(ing) for a certificate, or ( ) verify(ing) a digital signature" in Verisign's repository and that by doing the verification the user is agreeing to the terms of the agreement, including acknowledging that she has "access to sufficient information to ensure that (she) can make an informed decision as to the extent to which (she) will chose (sic) to rely on the information in a certificate."

The relying party is supposedly bound by the agreement which affirms that she has enough information to decide to what extent she will rely on the information in a certificate, and also that she is solely responsible for deciding whether or not to rely on the information in the certificate. In other words Verisign is making no statements about what the information in the certificate represents and instead shifts the burden to the relying party to make the download decision without providing them with the necessary tools and resources.

There are at least two flaws with this approach:

> It presupposes the relying party can agree that sufficient
> information will be on the certificate to make the
> determination as to whether she will rely on it or not,
> without having seen the publisher's Web site.

The relying party must be able to receive authentication of a subscriber's public-key from a trusted-third-party (TTP) or the entire model is useless.

8. FACTUAL EXAMPLE OF FLAWS IN THE AUTHENTICODE SOLUTION

(a) Unforeseen Interactions

Consider two ActiveX controls. One provides a control similar to the Win95 "Start" button with all the commands on the user's computer presented in a list to choose from. Suppose it keeps these command names in a preferences file such as C:\windows\commands. The file may contain a list such as: Word, Excel, format c:, IE3, etc.

Consider a second ActiveX control that performs certain "housekeeping" functions on the PC at regular intervals. It automatically wakes up at a specified time and executes a list of commands such as backup, defrag, etc. Suppose it keeps its list of commands in, for instance C:\windows\commands. At the next scheduled interval the second control dutifully finds the file written by the first one and fires up Word, Excel, and then formats the C drive. Commands after this one are of diminishing consequence.

The user's hard disk is wiped clean and so are the "fingerprints" for Authenticode. Even if they are somehow located, who should the user point the law enforcement people towards? Both controls did exactly what they were designed to do, exactly what they advertised to do. Who is the user going to sue? Obviously neither "misbehaved." What happened was an unforeseen interaction between the two, and was only possible because ActiveX is given unrestricted access to those system-level tasks. With only a bit of planning it would be possible to come up with a cooperating gang of ActiveX controls to do deliberate theft via collusion where each program is only doing what it's "supposed" to, yet the total of their activity is much greater than the sum of the parts. Current methods of tracking events through logfiles are unable to accurately reflect the non-linearity that is clearly at work here in the interaction of the components. The only way to avoid this would be to strictly de-couple the controls, by not allowing any to share information with the other, such as giving each its own private file-space to write in. Although this is the approach used by Java's sandbox, alas it is not possible in the "security-free" world of ActiveX.

(b) Proving the Origin of the Malicious Code Can be Almost Impossible

In the event the malicious code does not either reformat the user's hard disk or destroy its digital certificate outright there is still a great deal of uncertainty as to how the particular malicious code at fault can be identified as the cause of any particular harm. Certainly it would be easy if the damage occurred immediately after the ActiveX control was downloaded. But if it does something indirect; or waits until executed the 100th time; or modifies some other program so that it later does something nasty; then tracking down the source of the original corruption will be extremely difficult.

Assume for example that a component is signed by the real author, who was certified by a competent CA to be a reputable softwaie developer. The user reviews the certificate at install time, and accepts it on the basis of the reputation of the developer. The user then forgets about the code for some weeks to come. Later on, he or she visits a page of a hacker, or a page of a web site that has been broken into by a hacker, and the IE browser invokes the code with arguments supplied by the hacker. The code may appear to do what it's supposed to, or appear to do nothing at all while it's erasing the web browser's history file. The user may not even be aware that code is executing. The user goes on to about 50 other Web pages that night, and shuts off their machine with no evidence of a problem. When they reboot they may have a huge problem, depending on what the code was reprogrammed to do. The Authenticode scenario suggests that the user can now call their lawyer to sue someone, but who do they sue? The hacker that the FBI can't track? The well intentioned but pressured software developer who wrote the harmless control that was manipulated by the hacker to cause the damage? The certification authorities like Verisign that have forty page disclaimers of liability? And even if someone could be sued, is this an acceptable remedy for being without their computer system?

(c) No Consideration is Given to the Author's Competence as a Programmer

In cases where a program such as ActiveX has the ability to act on untrusted data, it isn't valid to make a judgment of its security simply on the basis of trusting that the writer of the program is not malicious. Consideration of how competent they are at writing "safe programs" is also important. Users of ActiveX are being encouraged to accept or reject controls based on whether they think the signer is trustworthy or not. No consideration is given to the stronger, and more relevant criterion of the author's competence as a programmer.

Because third parties can provide potentially hostile input to Active X controls -- at least for those classified as "safe for initialization" -- the "appropriate diligence" for such a control is much greater than that required for an ordinary application. Even though a well intentioned author creates a "safe" program, unless it has been written using the appropriate security safeguards it can be made to cause damage through the actions of another ActiveX control.

(d) Microsoft Justifies the Inherent Security Risks of ActiveX by Arguing that Users Want and Demand a Rich Computing Experience.

It has been argued that the Java sandbox approach is too restrictive, and that users want and demand a rich computing experience. This may be true, but these same users would prefer to use the name of their favorite movie star or basketball player as a password. It is up to the computer professionals to maintain a balance between adequate security protection and ease of use. Users should be encouraged to take informed risks, but they must be given the guidance and tools to accurately perform the risk/benefit analysis. Authenticode deters users from taking informed risks because it fails to provide them with the information needed to make an informed decision while at the same time assuring them that it is at their disposal.

(e) The Myth That Commercial Software Publishers and Others Will Be Deterred From Writing and Distributing Malicious Software Because of the Potential Risks of Economic Loss and Legal Liability

Historically hefty financial barriers to entry into the software development market using traditional distribution channels have restricted the number of market entrants. However the Internet provides a very low entry-cost distribution mechanism that is not without an increase in associated risks. Lowering the entry cost increases the potential for abuse. Furthermore, automating the process increases the chance that the abuse may go unnoticed. No longer can it be assumed that software developers will not risk loss of their potentially small financial investment by loading malicious controls onto the Web that, if undetected, would serve their ends.

(f) Average User Lacks the Training and Resources Necessary to Make Appropriate Downloading Decision Based on Information Provided by Developer's Web Site

The average user is probably only able to recognize a handful of big name Internet-related software development companies and even fewer companies that develop ActiveX components. And yet users are being asked to decide whether or not they should download a particular company's ActiveX component based on whether they are "known" (which, according to Microsoft's definition means "trustworthy"). Assuming the developer is "unknown" to them, the user has no idea what information on the developer's Web site is needed to making this critical decision and yet Microsoft clearly states that the user "can make the decision on how to run the code" based on the information provided in the certificate.

Furthermore, the average user will probably be reluctant to spend much time seriously evaluating the trustworthiness of a software developer and will instead base their decision on the site's professional appearance or some other intangible and possibly irrelevant factor. According to Michael Sullivan-Trainor, director of International Data Corp.'s Internet program, "The problem with the Web is that the sleaziest company in the world can put up a site as slick as the most respected corporation. Shopping (and downloading software) on the Web requires a little more investigation." Because a professional appearance can easily be created by the most criminal of software developer's it cannot be used as a measure of the developer's trustworthiness and yet Microsoft provides no guidelines to assist the user in making this analysis. Nevertheless they continue to assert, as stated above, that "the value of this type of certificate is in the information it provides to the user so that he/she can make the decision on how to run the code" and that this should be "comforting information."

(g) Contrary to Microsoft's Claim, Downloading Software From "Known" Software Vendors Does Not Necessarily Eliminate Risk

Implicit in the Authenticode trust model is the belief that all ActiveX components created by "known" software developers will be harmless and can therefore be trusted and downloaded without reservation. The recent track records of several software developers including Microsoft, seriously undermine this notion. According to an article called "Microsoft Security Flaws Run Deep," in the March 6, 1997 issue of CNET's NEWS.COM authors Nick Wingfield and Alex Lash state that "ActiveX is not the only security headache Microsoft is suffering. There are problems with its Internet Explorer browser." The article goes on to explain how earlier that week a group of students (does not specify whether they were students from the local university) found that by planting "Shortcuts" on a Web site they could trigger resident Windows 95 and NT programs to delete and manipulate files on a user's computer when browsing the Web site. According to the article Microsoft developers worked around the clock to fix the security hole.

In response to this IE "Shortcuts" security hole Stephen Cobb, director of special projects at the National Computer Security Association (NCSA) states, "I would say that you have to seriously question the integrity of Internet Explorer at this point because this was such a big hole." Cobb goes on to comment that "Microsoft's statement that they did a lot of testing (on Internet Explorer) is worrying, because if they did a lot of testing and didn't find this problem, their testing is very flawed." In all fairness, it must be pointed out that security holes are being found in other software developer's products as well, however the significance of Microsoft's track record in this particular case is that they are the ones that are making the argument that if the software developer is "known" then their ActiveX components must be trustworthy, and that the only criteria that is important is whether or not the user recognizes the software developer.

The same CNET article also points out that even if no one's computer is actually damaged by a security hole that is subsequently discovered after the user has downloaded software, individuals and companies still have to spend time and money to install the security patches on their systems. Stephen Cobb concludes that "(I)t's difficult for Microsoft to weasel its way out with the 'it does no damage' excuse, because (in the case of the "Shortcuts" bug) systems administrators are already looking at a big cost hit."

There is no empirical evidence to support Microsoft's assertion that downloading software from "known" origins is less risky than from "unknown" sites. Nor does this assertion take into consideration the possibility of a hacker placing a malicious control on a "known" Web site, or the possibility of a hacker "spoofing" a "known" Web site. Either of these can be done without detection either by the user or by the Authenticode system.

Joel McNamara explores this same issue in the June 1997 issue of Infosecurity News. In an article titled, "Security-Market Dynamics" he writes, "As security professionals, we like to think that security ranks right up there on everyone's most-important list. But when security isn't the primary purpose of the product, security features all too often take a back seat." McNamara lists some of the security holes that have been discovered recently in many of Microsoft's products ranging from Windows NT, Windows 95, WORD macro viruses, to Internet Explorer, Authenticode and ActiveX. Joel observes that "Microsoft's testing methodology appears to be more oriented toward discovering classic, show-stopping bugs rather than searching for more subtle, exploitable security holes." He concludes that, "(i)f people continue to buy products with marginal security, why spend the extra time and money implementing high-end security. . . . Unfortunately, the marketplace usually needs to yell, scream and threaten to walk away before it gets what it wants. So, until then, expect to see security as little more than just another check on a marketing features list." A user can be exposed to significant security risks even when downloading software from a "known" developer such as Microsoft.

(h) Relevance of Authenticode "Trust-Model" for users outside the United States

> Software developers located outside the United States but who wish to allow their components to be downloaded in the U.S.

According to the Verisign Web page, "Digital Ids for Servers: High-level Security at a Low Cost:"

> If your company has a Dun & Bradstreet (DUNS) number, you can complete your Digital ID request online. If you do not wish to use a DUNS number, or your company is not in the US, you can complete the enrollment form electronically and fax or mail Verisign any of the following pieces of documentation to establish your company's identity:
>
> - Articles of Incorporation
> - Partnership Papers
> - Business License
> - Fictitious Business License
> - Federal Tax ID Confirmation

Even assuming, for the sake of discussion, that Verisign's document authenticator's are familiar with the Articles of Incorporation or foreign equivalent for every country, and is able to make a reasonable effort to detect a faxed fraudulent document, how will the user who relies on the Digital ID know whether that foreign country even has any laws that will allow him some measure of recourse in the event that he suffers injury caused by the developer's software?

> Software developers located outside the United States but who wish to allow their components to be downloaded both in the U.S. and overseas.

Verisign has begun "franchising" overseas Certification Authorities who wish to base their practice statements on the Verisign "Certification Practice Statement" (CPS). Although several are under development, BelSign (www.belsign.be) is the first franchisee to go productional, , and their stated territory is limited to Belgium and Luxembourg.

So far little details are available about identity authentication procedures and other practical considerations and responses to e-mail inquiries have not been forthcoming.

(i) Web sites Can Be Spoofed or Hacked

In December, 1996 the Secure Internet Programming team at Princeton University published a technical report describing an Internet security attack called "Web spoofing." In this scenario, an attacker:

- Creates a shadow copy of a web page;
- Then, funnels all access to the web page through the attacker's machine;
- And finally, tricks the unwary consumer into revealing sensitive or private data, such as PIN numbers, credit card numbers or bank account numbers

Web spoofing requires that the attacker be able to interject his machine between the server and client, in a man-in-the-middle attack. Although under some situations certain visual cues may be used to detect the presence of a spoofed Web page, these can be eliminated by the skilled programmer. The only real solution is to check the "View Source" option and read the html source code for the Web page the user is currently browsing to know for certain whether their browser is connected to the correct site. Even a server and client using SSL can be spoofed if the hacker is able to intercept the client's initial request for authentication to the server and before a secure link is established.

Once the unsuspecting user is connected to the attacker's bogus Web page, all transactions between the user and the certification authority can be intercepted and fraudulently manipulated. Thus, a harmful ActiveX program could easily be made to look as though it came from a "known" and trustworthy developer. After the program has downloaded to the user's PC and done its damage there is no way for the user to identify the developer because the program never had a Digital ID in the first place. Furthermore, the knowledgeable hacker will delete or modify the browser's history file so no record would remain of the user's visit to the spoofed Web site.

According to Ed Felten, co-founder of the Princeton Internet Programming research team, there have been reports of the FBI investigating false sites and forcing them to shut down and then charging them with wire fraud. Felten believes that "(a)s the stakes increase, there is a chance for it to happen more and more."

(j) Obtaining a Digital ID Through Fraudulent Means

Fred Mclain, software developer, consultant, and author of the now infamous ActiveX "Exploder" control (see below), provides the following perspective on the Authenticode "code signing" process, from a FAQ on his personal Web site located at www.halcyon.com/mclain/.

> Code Signing simply attempts to identify who signed the control. Anyone can go out and get a code signature. It's a pretty much automatic process. You go to a web site, give them a name, address, credit card number and some other stuff (none of which have to be yours), click "I Agree" on a page full of legal jargon, and pretty soon you get an e-mail with the information you need to sign the control in it. Once you have your Digital ID, you can sign any unsigned ActiveX control. Nobody reviews these controls! In other words, a signature doesn't tell you who wrote the control and it doesn't tell you if the control is safe or not. Heck, with the number of hot credit card numbers out on the net, it doesn't even tell you for sure who signed it. A danger is that seeing that a control is signed will give folks a warm fuzzy feeling about the control, and encourage them to run it, even though it does not guarantee their safety!

A recent Associated Press news item from San Francisco dated May 22, 1997 demonstrates the prevalence of credit card theft on the Internet and the accessibility to those stolen numbers. The article reports that according to Bureau spokesman George Grotz, the FBI recently arrested a hacker who used a "sniffer" program to eavesdrop on electronic transactions between customers and a dozen companies selling products through a major Internet provider. The sniffer software gathered 100,000 credit card numbers along with enough information to use them. The hacker was arrested for allegedly attempting to sell the information to an undercover FBI agent who saw the hacker's advertisement on a computer bulletin board.

FBI statistics indicate that the majority of computer crimes go undetected, and, until recently, most of the ones that are detected are never reported. Therefore it is safe to assume that there are many other sources of fraudulent credit card information gathered from the Internet that are available to persons registering ActiveX controls. Frequently the credit card owner will not realize their number has been stolen for several weeks or months, depending on the thief's spending patterns. As a result, if a stolen credit card is used to acquire a Digital ID using fake identification, the fraudulent charges will go through undetected and because there is no retroactive follow-up on the part of Verisign or Microsoft, the certificate will remain valid even after the card theft has been discovered and the card invalidated, unless the defrauded consumer makes the effort to contact them which is unlikely.

FACTUAL EXAMPLES OF ACTIVEX-RELATED SECURITY RISKS

(1) InfoSpace Program Compromises Authenticode Security

On September 23, 1996 CNET-Online and other publications reported that Lycos, a WWW Search engine company posted a program on its Web site that would allow downloadable programs with InfoSpace Digital Ids to bypass the Authenticode security controls in Internet Explorer.

Nick Wingfield's article "Program compromises IE security" explains that because the program which was created for Lycos by InfoSpace, a startup Internet company, circumvents IE's security warning window, InfoSpace could sneak programs onto a user's personal computer without warning.

InfoSpace executives denied that there was any malice intended in its program, adding that it has provided Lycos with an updated version of the code. Lycos planned to post the new program later that evening, according to InfoSpace. "It was a bug that got incorporated into the production code," InfoSpace CEO Naveen Jain said.

Although the InfoSpace program apparently was not created with malicious intent, according to Wingfield "it underscores the fragility of Internet Explorer's security defenses, as well as broader security issues related to downloading over the Internet."

"Code signing is not a guarantee of code quality," Charles Fitzgerald, a product manager at Microsoft said. "It's an accountability trail."

The InfoSpace 'bug' modified the Windows 95 Registry configuration setting by simply registering InfoSpace as a "Trusted Publisher" thereby allowing all code from InfoSpace to be downloaded automatically without requesting the user's consent. The operation is akin to inviting a guest over to your house for dinner before you leave town for a month-long vacation and having them copy the key to your front door without permission. If the guest enters your house while you're gone and a neighbor questions him about it, the guest only has to show the neighbor the copy of the key as confirmation he has your permission to enter. Whenever the user's browser detects an InfoSpace program it will automatically be downloaded without the user's awareness or consent, because Authenticode has been told to automatically trust all InfoSpace developed programs.

"Clearly their software is doing something a tad aggressive," said Rob Price, a group program manager for Internet security at Microsoft." (With Authenticode), users are making a one-time trust decision, this is a persistent trust decision."

(2) Symantec Corporation's Norton Utilities Victimized by Malicious ActiveX Control

According to information posted on their Web site (www.symantec.com), on April 7, 1997, Symantec was notified that a malicious Web site had been created that uses an ActiveX control to gain access to a user's PC if they use Norton Utilities 2.0 for Windows95 and get on the World Wide Web. Because a specific component (TUNEOCX.OCX) of the Norton Utilities System Genie is marked as a script file, ActiveX-aware WWW scripts can make use of it as an ActiveX control. The result is that a malicious user could use the script to run any command, such as delete, format or ftp, on the local host. Symantec responded to the news quickly and responsibly, posting a fix for the problem within 24 hours.

(3) "Exploder" Control

Software developer and consultant Fred Mclain created a live demonstration of ActiveX's capabilities in late summer of 1996. Mclain created an ActiveX control which he called "Exploder" and which he placed on his Web site with the explanation that it would perform an automatic "graceful" shutdown of any user's PC running Windows95 who chose to voluntarily click on the control link and automatically download it to their PC. Because the control caused a "graceful" shutdown no damage was caused to the user's PC, but the damage to Microsoft's image was immediate and irreversible. As recently as April 1997, Sun Microsystems CEO Scott McNealy was still demonstrating MClain's Exploder control to crowds of Java enthusiasts.

(4) Germany's Chaos Computer Club Live Demonstration To Make Bogus Money Transfers From Intuit's Quicken Online Banking Customers

The Chaos Computer Club (CCC), a German hackers group from Hamburg, demonstrated on national TV in February 1997 that they can use an ActiveX control to steal money from one account and put it into another without the use of a Personal Identification Number (PIN) during an online banking transaction.

CCC showed that once the ActiveX control is downloaded by a user browsing their Web site who uses Intuit's Quicken for electronic banking, the control will add an extra electronic fund transfer command to the pending transfer list. The next time the user does his or her banking online, the bogus transaction will get executed along with the rest without alerting the user.

The Computer Club's stated purpose in holding this public demonstration was to alert people about the risks associated with doing business on the Internet and specifically with ActiveX.

Intuit, the company that develops Quicken, responded by recommending that users disable the ActiveX controls in their Internet Explorer browsers or switch to the Netscape Navigator if they are concerned about the safety of ActiveX controls. The company also stated that of the 9 million copies of Quicken currently in use worldwide, the present U.S. version of Quicken can only be used to transfer money from "pre-authorized" accounts as approved by the user. A future German version of the software will have encryption features to prevent hackers from breaking in. To its credit, Intuit did an excellent job of public relations "damage control" and used wide,the Web, because it is the the situation as an opportunity to educate consumers on how to take proper safeguards to protect themselves on the Internet in general and from similar situations in the future.

RECENT MICROSOFT SECURITY ENHANCEMENTS

(1) Microsoft's Authenticode 2.0 - Band-Aid for a severed artery

Microsoft recently announced Authenticode 2.0, a significant upgrade to the initial version which was first released less than one year ago. On the plus side the new upgrade includes a number of features Microsoft says will make downloading code safer, including time-stamping support to ensure that code was signed with a valid digital certificate. Various Microsoft bulletins and announcements inconsistently report that it also supports access to certificate revocation lists (CRLs), a feature that checks in with an online list of revoked certificates before downloading code.

*However, on the negative side the logistics of the upgrade are cumbersome, time-*consuming and will potentially result in delays while unsuspecting users are forced at the last minute to download either the upgrade. Software publishers who have signed their code prior to June 1997 must re-sign their code by June 30, or before their current Digital ID expires. According to Microsoft, because Authenticode 2.0 checks the revocation list to determine whether the Digital ID is still valid, it will notify a user who wants to download an control that has not been re-signed as either unsafe to download (if their security is set to High), or out-of-date (if their security is set to Medium). Only code that has been re-signed will appear in the revocation list as safe to download.

This upgrade is significant as a validation of Microsoft's willingness to obfuscate the facts and fabricate its own reality, in its single-minded pursuit of market share. Prior to this upgrade a user was expected to navigate the maze of menus and options on the Verisign Web site to locate CRL information. No explanation or instructions were presented to the user when the subscriber's certificate appeared on their screen, informing him that he must inquire of this proprietary database to find out whether the Id used to sign the certificate he was viewing and potentially relying on was still valid or whether it was suspended or revoked. Also, without the time stamping capability, it was impossible for the user to tell whether the certificate appearing on his screen was signed using an expired Digital Id or not. Although Microsoft and Verisign engineered this upgrade prior to the time most Digital Ids and certificates would have expired, there was no advance acknowledgment of this limitation. One can only hope that other essential attributes of this ostensibly trustworthy Authenticode security model are not still on the drawing board to released later as enhancements.

(2) "Security Zones"

This new feature will let users or their network administrators arbitrarily divide the Web sites into four predefined zones: intranet, trusted extranet, general Internet and untrusted. Web sites can then be assigned to a particular zone, and be subject to the corresponding level of security protection. For example, ActiveX controls and Java applets coming from the Internet might be assigned to untrusted zones, and the administrator could prevent them from being downloaded by configuring that zones security protection accordingly.

In a sense this is just a 'macro' version of Java's 'sandbox' security model. The sandbox prevents Java applets from gaining access to sensitive system functions that are outside its boundaries. IE's security zones can also prevent Java and ActiveX programs from gaining access to sensitive system functions, depending on the way the security protections are configured. However, the user or administrator is unable to override or misconfigure Java's default sandbox protection, whereas the IE security zone protection can be turned off or improperly configured, leaving the user completely vulnerable.

THE FUTURE OF ACTIVEX AND DOWNLOADABLE AND EXECUTABLE CONTENT - Will it ever be safe to 'trust' again?

If Microsoft is unwilling, users must organize and develop alternative means of protecting themselves from ActiveX. Some examples of proposed alternatives include:

(1) Web of Distrust

One author is calling for an online, independent watchdog organization that "provides users with timely alerts on hazardous or questionable software." This group would act as a clearinghouse for reports of all harmful or suspicious downloadable and executable content. The information could be distributed by newsletters to subscribers, or available to any user by hyperlink access before they make the "fateful" decision to download. Kobielus writes, "Our best defense against malignant controls is to pool our experiences, expose the offending code-mongers to the entire online ". . . . Net community and thereby burn them out of existence."

Although certain legal issues and standards must be addressed before "burning" anyone out of existence, this approach could serve as a model for a more effective means of keeping Cyberspace free from harmful code.

(2) Better-Business-Bureau OnLine (BBBOnLine)

The Council of Better Business Bureaus, best know for their certification of local businesses in the physical world, have developed a new U.S. online service, "dedicated to helping consumers identify ethical marketers on the Internet and thereby make the Internet a safer, more reliable place to get information and conduct business." According to information on their Web site, companies that display an encrypted BBBOnLine CARE seal on their Web pages have demonstrated their commitment to a series of strict business standards for customer service and marketplace ethics. Consumers can hyperlink from the seal to the BBBOnLine home page to get a reliability report on the member company, including their management, time in business, relevant aspects of its goods and services, complaint experience and other evidence of responsible marketplace behavior. Several large corporations involved in Internet-related markets are co-sponsoring this service including, Hewlett-Packard, Xerox, Netscape, AT&T, and GTE.

Some examples of their rigorous Participant Standards include:

> Provide the BBB with information regarding company location, background, etc. which will be verified by the BBB in a visit to the company's physical premises;
>
> Be in business a minimum of one year (with limited exceptions);
>
> Respond promptly to all consumer complaints;
>
> Agree to binding arbitration, at the consumer's request, for unresolved disputes involving consumer products or services advertised or promoted online.

(3) PC-based Browser Add-on Security Products

Several vendors including Finjan Inc., and eSafe Technologies have recently released products that promise to provide protection against all Internet threats, whether they are hostile ActiveX controls or Java applets. eSafe Protect not only recognizes a set of known security holes and rogue controls, but it also has the ability to run in a learning mode. This allows the program to see where the user's browser and e-mail clients usually read or write data or execute other applications and develop a pattern of acceptable behavior (similar to an 'intelligent' sandbox model). After the learning period is completed (usually about one day), any activity outside of the normal range will generate an alarm, and require user intervention to proceed. As a result it also provides protection against yet-to-be-discovered security holes in popular Web browsers or other unknown hazards.

Independent Software Accrediter is Necessary to Determine Software "Harmlessness"

Digital signatures can measure the authenticity of a person, but not their intentions or competence. Until software developers see it is in their best interest to invest more resources into writing secure software a separate entity is needed to gather concrete evidence of the software developer's intention and competence in advance. By testing their software against industry benchmarks and providing guidance to the uninformed user interested in ascertaining the safety of the software they want to download this entity will bridge the gap between identity verification and a software publisher's intentions and competence.

The "Software Accrediter" will validate that an ActiveX component is both harmless and "safe" to operate in an "open" environment by testing it against a set of industry-wide programming and Internet security standards. For a control to be "harmless", it must be unable to cause damage by itself. For it to be "safe" the control must be designed and written with a level of programmer competence that prevents other controls from being able to advantage of programming flaws and force it to cause harm.

The Software Accrediter will take on significance in the use of downloadable and executable content to authenticate its conformity to the norms of programming and Internet security practice. For instance, where a Software Publisher Digital ID is executed and digitally signed by a Certification Authority, the "Software Accrediter" will issue a message of accreditation attached to the Digital ID which validates the harmlessness and safety of the program within certain parameters. The validation will identify the level of risk associated with the control and the user can make an informed decision whether or not to download the control, based on the potential injury he could suffer. Neither the mere application of a digital signature, or the restriction to "safe zones" satisfies accreditation requirements for these types of dangerous programs. The "Software Accrediter" will combine the benefits of digital signatures with industry-accepted software accreditation to provide high quality international control authentication in a measure far exceeding current practices.

Public key cryptography, or digital signatures, can be used to sign application software and certify it as "safe" as judged by some certifier, only if the software is held up against a set of industry standards - where one of the "safety" properties would be that the application cannot be corrupted by malicious external programs or data. Microsoft offers Authenticode as a way of empowering the user to determine whether individual downloadable executable Web content is safe to use. It purports to provide the user with information which will be "comforting" to them in their analysis. Unfortunately, Authenticode simply moves the burden of assurance on to the user, without making the analysis any more tractable. It places an unreasonable burden on users, who must decide which developers are trustworthy based on insufficient data and inadequate tools. Because even major mass market application software (e.g., Quicken) appears susceptible to attacks by malicious controls, it is not clear what this type of certification technique could add.

Netscape's Hybrid "Code-Signing" Solution

Netscape has recently released its own implementation of an Authenticode-like product that has much more robust security protection against harmful downloadable and executable programs. In addition to the generic characteristics of a digital signature; authentication, integrity and non-repudiation, "code-signing" also determines what an ActiveX control or Java applet wants to do on the user's machine. Netscape's Communicator checks to see if the software is signed and attempts to verify the signature. If the applet is unsigned or if the signature is unverified the applet is automatically restricted to running inside the "sandbox."

When the downloaded program wants to get access to a PCs system resources a dialog box is displayed that shows the user what kind of access it wants, the identity of the signer, and the associated risks. With this information the user then decides to allow or deny the access that the Java applet has requested.

ActiveX controls can be packaged in such a way as to fulfill the Java specifications necessary to allow code-signing. This process is accomplished using the JAR Packager tool which creates an envelope around the control that results in a cross-platform JAR file. The JAR Packager is a tool that allows developers to sign, envelope and compress Java applets, plugins, and any other type of file. The JAR file format was a joint effort between JavaSoft and Netscape.

In the future, an evolving combination of these and other approaches will be used to provide protection. Security guru Gary McGraw believes the long-term solution combines "code-signing authentication and some sort of security model, like a (Java) Sandbox." He believes it will be "much easier to (add code-signing) to extend Java, . . . than it will be reverse engineer Sandbox into ActiveX."

SUMMARY

The general outlook for ActiveX as a computer security problem is unclear. The potential vulnerabilities are legion. Bearing in mind the FBI's computer crime statistics indicate that over 80% of all detected computer crimes go unreported, and many more of them go undetected, during its initial 18 months in existence exploitation of ActiveX has been virtually non-existent. Unfortunately, as the economic incentive for creating malicious ActiveX controls increases, it seems likely that attackers will attempt to exploit its security vulnerabilities.

Given the obvious security risks presented by ActiveX, combined with the absence of broad-based support for Authenticode, the only possible explanation for Microsoft's continued pursuit of this folly is a last-ditch effort to keep its hand in the Internet game and maintain its share of the desktop computing software market. Microsoft is committed to maintaining its monopolistic hold on the PC and Internet software industry by marketing its auto-immune deficient ActiveX software product, and its parasitic partner Authenticode. Even with the intellectual horsepower at its disposal it appears to be unwilling to develop a secure alternative because then there would be little incentive for users to purchase its Internet Explorer Web browser, and there would be little hope for Bill Gates' vision of a single, seamless Windows-based PC desktop and Internet interface.

CONCLUSION

This article has presented some good points and bad points about ActiveX and Authenticode both of which have only been in existence for less than two years. It is inevitable that both security protection for downloadable and executable programs and Certification Authority policies and practices will evolve gradually. Nevertheless, in the interest of minimizing the risk exposure to the user, it would be prudent for software developers to acknowledge these risks up front and allow users to understand them and begin making informed decisions based on accurate information, or paying customers must demand something better. Risks associated with downloading any software from the Internet are unavoidable, but Microsoft chooses not to explain those risks to users or give them the tools to properly manage those risks. Instead what Microsoft does provide is confusing, contradictory FAQs, bulletins and marketing announcements that even go so far as to state, "Because Microsoft must respond to changing market conditions, this document should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication."

Microsoft understandably wants to be the first to market with each of its latest Internet software products so it can gain whatever advantage it can over its competitors. But they are cutting corners at the customer's expense by leaving necessary security features out and the customer needs to be informed to decide whether it is an acceptable expense. In the wake of Love Canal, Three-Mile Island, Hanford Nuclear Reactor, Rocky Flats and other life-threatening breaches of the public trust we have matured as a nation to the point where even the courts support our right to receive advance notice before toxic chemicals are pumped into our back yards and personal spaces. Yet Microsoft is allowing toxic ActiveX components to be downloaded into our PCs without reasonable notice and disclosure of all the risks by pretending that it's fake security system Authenticode can provide reasonable detection and defense.

The most effective long-term technical solutions appear to require systemic changes in the way computer software is built and the way software standards are developed and enforced. The safest near-term alternatives for the majority of users all involve giving up many of the "bells and whistles" that make Web browsing so entertaining by configuring Internet Explorer browsers to restrict all ActiveX controls from being downloaded to the desktop.

Microsoft recently announced on TechNet that, as of the release of XP, the only way that consumers and businesses can make on-line purchases, or submit private data (e.g., on-line banking) through a "secure" (SSL-enabled) Web site, is by using new features that are available exclusively on Windows XP, via the Windows Update Web site. Users of Microsoft NT, ME, and W2K may install an "upgrade patch" that will allow them to manually download new root certs, and to use a limited subset of the XP-based capability.

> To better protect Microsoft customers from security issues related to the use of public key infrastructure (PKI) certificates and enhance the experience for Windows users, Microsoft is moving to standardize and clarify the criteria for root certification authorities in Windows XP. This standard also applies to root certification authorities in Internet Explorer and any other Microsoft product.
> (http://www.microsoft.com/technet/security/news/rootcert.asp)

Let me repeat, as of the release of XP next week, the ability for consumers using non-Microsoft operating systems to perform "secure" transactions via Internet Explorer (IE) will be severely curtailed, and over the coming months, entirely eliminated.

> When a user visits a secure Web site (that is, by using HTTPS), reads a secure e-mail (that is, S/MIME), or downloads an ActiveX control that uses a new root certificate, the Windows XP certificate chain verification software checks the appropriate Windows Update location and downloads the necessary root certificate. To the user, the experience is seamless. The user does not see any security dialog boxes or warnings. The download happens automatically, behind the scenes.

Microsoft has no plans to provide an "upgrade patch" for the non-Microsoft versions of IE that it currently supports (e.g., Solaris, Linux, HP-UX, and Mac.). Microsoft properly considers Auto Root Update and Windows Update to be *Windows technologies* for conveniently keeping users up to date with certificates in the Microsoft Root Program (the user doesn't have to take many steps to install the roots). However, it has no plans to provide these convenience mechanisms for non-Windows platforms at this time.

The result is that the only way that CAs or on-line merchants can get their certificates into the IE browsers of non-Microsoft consumers is by forcing the consumer to manually download and install the certificate directly from a Web site. This eliminates any level of trust assurance that may have resulted from IE's existing root certificate accreditation process.

Under this new regime, when a consumer using IE on a non-Microsoft platform enters a secure Web site to make a secure on-line purchase, he is prompted to download and trust the CA root certificate of any merchant whose root is not already in that browser. The same is true if a Web site wants to download an ActiveX control, which is signed by an unknown and hence "un-trusted" Publisher.

Eliminating future access to new root certificates in its IE browser will deprive consumers using non-Microsoft platforms from the ability to conveniently and "securely" make purchases at a secure Web site (HTTPS), read secure e-mail (S/MIME), or download signed ActiveX controls with the same level of trust assurance that he experienced prior to this new regime.

This change will adversely affect the consumer, the on-line merchant, and the CA, as each of them has a stake in making the on-line experience as smooth, secure, and convenient as possible. This latest manipulation of the Internet software market by Microsoft will provide consumers with a strong incentive to migrate to a Windows platform, so they can continue to use the Web with the same degree of ease, and sense of security as before.

In addition, some commercial PKI applications and products are designed around consumer access to their root certificates in Microsoft's IE. Eliminating consumer access to their root certificates from IE will force them to restructure their applications, and in some cases their whole product strategy. Of course, Microsoft will argue that these vendors were receiving a "free ride," while it developed the technology to tighten up its PKI solution. However, Microsoft's PKI solution is anything but "tight," and in fact, it is still quite immature. In addition, it will remain so for several years, to the detriment of the consumer, and the industry.

This tactic is virtually identical to the one that Microsoft used to eliminate competition in the browser market. It offered features similar to Netscape's, but at no charge, because it could afford to use its income from OS sales to offset the loss it took on its browser product. Initially, Microsoft's browser was inferior to Netscape. However, over time, as the marketing power of the Windows desktop gradually surmounted Netscape's marketing channels, and as Microsoft commandeered many of the existing Internet browser standards, IE achieved a superior market position.

This time Microsoft provided consumers and the industry with "free" access to CA root certificates embedded in IE. However, now that it believes it has eliminated any competition for this service, Microsoft intends to force consumers to purchase XP or another Windows platform, so they can continue to enjoy the same convenience and benefits from digital certificates as before.

Although Microsoft will certainly claim otherwise, I believe it is well within its power to continue to support the storage of new root certificates in non-Microsoft versions of IE. However, Microsoft representatives have indicated that they have no plans to do so at this time. As are result, consumer trust in on-line commerce, and the viability of many PKI solution vendors will both suffer in Microsoft's latest grab for another piece of the Internet software market, PKI. Microsoft's PKI solution is inferior to current alternatives, and it will not achieve its promised capabilities for many years, after using the public as its testing ground.

Is Microsoft trying to corner another piece of the Internet software market by illegally leveraging its market powers, as the court agreed that it has done in the past? The pattern is virtually identical.

Rick N. Hornbeck
556 S. Fair Oaks Ave., Suite 346
Pasadena, CA 91105
Rick_Hornbeck@pacbell.net
(323) 363-2151

PROFESSIONAL EXPERIENCE:

HORNBECK CONSULTING – Pasadena, California 2000 – Present.

- Security Policy, Certificate Policy, and Certification Practice Statement consulting and development;
- Internet and network security policy consulting;
- PKI legal issue spotting and consulting. Representative topics include privacy; identity authentication; "qualified" certificates, security services, jurisdiction, and digital and electronic signatures; and local, national, and international regulations and case law in both civil and common law jurisdictions.

Customers include:

- **ENSURELINK CERTIFICATION AUTHORITY** – San Diego, California 2000 - Present. *PKI Consultant* – Certificate Policy, Certification Practice Statement, and PKI-related consulting.
- **ALPHATRUST CERTIFICATION AUTHORITY** – Dallas, Texas 2000. *PKI Legal Consultant* – Certificate Policy, Certification Practice Statement, and PKI-related legal consulting.
- **EXPERIAN** – Orange, California 2000. *PKI Legal Consultant* – Consulted with in-house legal counsel, defining and documenting application-specific, PKI-related legal issues.

ENTRUST TECHNOLOGIES - Plano, Texas 1999 - 2000. *Senior Security Consultant* - Developed Security Policies, Certificate Policies, and Certification Practice Statements for large national, multi-national, and international organizations. Worked directly with senior client management to determine their PKI requirements. Worked with sales force on national opportunities. Worked with consulting partners to out source PKI consulting work during peak periods and on joint projects. Provided on-site classroom training programs lasting 3 – 4 days for consulting partners and customers on Entrust-specific security and PKI consulting methodology and concepts.

Customers include: Experian, Bell Atlantic, MCI WorldCom, Hoffman-LaRoche, State Farm Insurance, First American Real Estate Information Services (FAREIS), Ernst & Young, Price-Waterhouse Coopers, People's Bank of China, Capital One, US Department of Agriculture, Fidelity Investments, Illinois Secretary of State, First Data Corporation, ...

OFFICE OF COURT ADMINISTRATION, STATE OF TEXAS - Austin, Texas 1997 - 1998. *Strategic Technology Planner* - Responsible for the implementation of a statewide computer and communication network linking all state courts. Developed supporting rules, policies, guidelines, and statutes relating to the electronic filing of court documents. Prepared cost analysis and preliminary design for the Texas Judicial Committee on Information Technology, based on planned technology.

ELECTRONIC COMMERCE SYSTEMS – Los Angeles, CA (1995 – 96); Austin, TX 1997 – 1998. *Principal* – Consulting company provided electronic commerce consulting services with an emphasis on Internet and Web-based security, public-key infrastructure (PKI), digital signatures, electronic filing of court records, and electronic payments.

Customers included:

- **Wells Fargo Bank** - Los Angeles, California, 1996.
  Database Developer - Designed and developed MS-ACCESS database integrating First Interstate Bank commercial loan database with Wells Fargo data following bank merger.
- **Orange County Superior Court** *(intern, part-time)* - Santa Ana, California, 1996.
  - o **Court Technology Department** - Drafted new court rules for electronic filing of pleadings via the Internet for pilot family law electronic filing project.
  - o **Law and Motion Research Department** – Reviewed, researched, and summarized legal motions for judge's Law and Motion hearings.

LAX SHUTTLE TRANSPORTATION CONSORTIUM, El Segundo, California 1996. *Arbitration Hearing Officer (part-time)* – Arbitrated appeals from personnel disciplinary actions.

ATTORNEY GENERAL'S OFFICE, DEPARTMENT OF JUSTICE, STATE OF CALIFORNIA - Los Angeles, California 1995 – 96. *Legal Intern (part-time)* - Wrote briefs, motions, and memos; performed legal research in support of Deputy Attorneys General; assisted in trial preparation.

COMPUTER SCIENCES CORPORATION - El Segundo, California 1993 - 95. *Senior Management Consultant* - Developed Information Systems Strategic Plan and Architecture for United States Air Force, Materiel Systems Command, Los Angeles Air Force Base (LAAFB). Delivered an integrated, base-wide strategic plan encompassing reengineered business processes, network operating systems, e-mail, and network security for over 25 unique, on-base Air Force organizations with disparate computer and network platforms.

TRW SPACE & DEFENSE (ELECTRONIC SYSTEMS GROUP), Redondo Beach, CA. 1988 – 93. *Network Systems Engineer* - Led team in design, development, and implementation of a reengineered purchase order processing system using state-of-the-art client-server technology linked with the corporate network. Implemented software upgrade for Procurement EDI application and integrated with batch FTP transfer from mainframe.

Responsible for implementation, administration, and security of multiple, inter-connected local area network servers running SCO UNIX, AT&T System V.4, and SUN OS over TCP/IP, and DOS/Windows clients.

PRICE WATERHOUSE, OFFICE OF GOVERNMENT SERVICES, Los Angeles, California 1987. *Senior Consultant* - Created functional model for reengineered application in support of 'Los Angeles Employees Retirement Association' (LACERA) software development project team.

XEROX CORPORATION, PRINTING SYSTEMS DIVISION, El Segundo, California 1985 – 87. *Senior Analyst/Programmer* – Supervised two analyst/programmers and coordinated design, development and implementation of purchase order entry system for Printing Systems Division.

TRANSAMERICA CORPORATION, Los Angeles, California 1983 – 85. *Analyst/Programmer* – Assisted in design, development, and implementation of a nation-wide information system enabling insurance agents to submit customer applications for insurance coverage directly into the mainframe computer in the home office from field offices across the country.

TEACHING EXPERIENCE:

- **UNIVERSITY OF PHOENIX ON-LINE,** 2001
  Part-time instructor – Risk Management in a CIS Environment (Computer Security); Contracts, Ethics, and Intellectual Property;
- **SANTA MONICA COLLEGE,** 2001.
  Part-time instructor – Introduction to Computer Systems;

- **CALIFORNIA STATE UNIVERSITY, LOS ANGELES,** 2001.
  Part-time instructor - Internet security.

EDUCATION:

LOYOLA LAW SCHOOL, Los Angeles, California.
Juris Doctor - December 1996.
Dean's List Honors, 1995.
California State Bar Foundation – Public Service Grant 1996.

UNIVERSITY OF SOUTHERN CALIFORNIA, Los Angeles, California.
Master of Science, Information Systems Management - May 1990.

CALIFORNIA STATE UNIVERSITY, LOS ANGELES, California.
Bachelor of Science in Business, Minor in Business Information Systems - June 1983.
Dean's List Honors, 1982.

ADMITTED TO PRACTICE

State Bar of Utah - May 2000, Active Member.

ARTICLES, STANDARDS ACTIVITY, PRESENTATIONS, AND COURSES TAUGHT:

PUBLISHED ARTICLES:

- *Electronic Filing of Court Records: Standards and Open Systems* (West Group 1998);
- *Electronic Court Filings for Attorneys: What, Where, When, Why and How* (West Group 1998);
- *Into the Breach: Understanding Security Issues Involved in Commerce on the Internet - Parts I and II*, The DataLaw Report, (Clark, Boardman, and Callaghan 1997);
- *The Troubling Truth About "Trust" on the Internet*, Journal of Electronic Commerce, (EDI Group, Ltd. 1997);

COMPUTER SECURITY, PKI STANDARDS, AND RELATED ACTIVITY:

- Internet Engineering Task Force (IETF) "RFC 2527," *Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework*, attributed contributor (March 1999);
- *GUIDeS – Guidelines, Methodologies and Standards to set up a CA for Digital Signatures*, European Commission, attributed contributor (June 2000);
- American Bar Association (ABA), Information Security Committee, *Digital Signature Guidelines*, drafter (August 1996).
- High-Technology Crime Investigation Association – Southern California Chapter, Member.
- Internet Corporation for Assigned Names and Numbers (ICANN) – Member At Large.

PRESENTATIONS AND COURSES TAUGHT:

- *Risk Management in a CIS Environment*, University of Phoenix On-line, July-August 2001.
- *Certificate Policies and Certification Practice Statements in a Network Trust Model*, The Internet Security Conference (TISC), October 1999 Boston, MA;
- *Electronic Filing of Court Records: Standards and Open Systems*, American Bar Association Annual Meeting, Presidential CLE 1998.
- *Electronic Filing of Court Records: A Conceptual Framework*, 1998 ABA TechShow;
- *Introduction and Intermediate Public-key Infrastructure (PKI); Digital Signature, and Related Standards at the State, Federal, and International Levels; Certificate Policies and Certification Practice Statements* (Entrust) 1999;
- *Introduction to UNIX Operating System*, San Jacinto Community College, Clear Lake Texas, (NASA) 1997.

FOREIGN LANGUAGES: French (Fluent), Spanish (Proficient).